* 1. Practising Recurrences

1.T(n) = 6T(n/3) + Θ(n^2)

Master Theorem:

Θ(n^2) -> Non- Recursive and Asymptotically positive

a = 6

b = 3

By Case 2 : T(n) = Θ(n^2)

2. T(n) = 5T(n/4) + Θ(n^2)

By Master Theorem case 2 : T(n) = Θ(n^2)

3. T(n) = 3T(n/2) + Θ(n √n)

By Master Theorem case 2 : Θ(n √n)

4. T(n) = 4T(n/2) + Θ(n^2)

By Master Theorem case 2 : Θ(n^2logn)

* 1. Anagram detection
* Get string s1
* Get String s2
* Perform Merge sort on s1 and then on s2
* Now compare two strings to see whether they are equal

Time Complexity = Merge sort on s1 + Merge sort on s2 + Comparing s1 and s2 which is O(nlogn)+O(nlogn)+O(n) asymptotically equals O(nlogn)

By choosing quicksort with randomization in selecting pivot, space complexity also gets reduced